technische universität
dortmund

Presentation: Active learning - minimising expected error and variance

Sathish Kabatkar Ravindranth

January 28, 2024

TU Dortmund University - Department of Statistics

Example: Simple classification scenario

- A poultry farmer grows goats, chickens, and turkeys and harvests fruits on a small plot of land.
- The farmer uses the fruits to feed the animals, but the safety of the fruit affects the animal's health.
- The veterinarian suggests that animals eat smooth fruits because they are healthy, and who eats the irregular fruits tend to fall ill.
- The farmer has use of both types of fruits. If the fruits are safe, he can feed animals; if the fruits are not safe; he can decay them and use them as fertilizers. So, the farmer wants to distinguish between safe and noxious fruits as accurately as possible.
- The veterinarian says the shape of a fruit is the only feature related to its safety.

Example: Defining the classification problem

Assume the "irregularity" is quantified as a continuous metric. Now, classify the fruit either noxious or safe based on the shape of the fruit.

- One feature: irregularity measure (i.e. measure on shape) continuous $\mathcal{X} : x \in \mathbb{R}$.
- Binary response: Fruit is safe or noxious $\mathcal{Y} : \{safe, noxious\}$.
- One of the most important goal of classification is to achieve high accuracy.
- A simple classifier function $h : \mathcal{X} \to \mathcal{Y}$, parameterized

$$h(x; \theta) = \begin{cases} \oplus \text{ safe} & \text{if } x < \theta, \text{ and} \\ \ominus \text{ noxious} & \text{otherwise.} \end{cases}$$

- The main challenge here is to find a optimal threshold θ^* where fruit become noxious from being safe.

Motivation: Trivial solution

- Using a supervised learning technique. Traditional way to estimate θ^* from a set of labeled instances $\mathcal{L} :< x, y >$, x is the continuous feature and y is the binary label, either "safe" or "noxious".
- Set of labeled data $\mathcal{L}$ is collected by having animals eat the fruit and observing their reaction.
- We assume that every animal reacts in the same way.
- Arrange a large number of fruits based on irregularity measure x.
- Test all fruits and collect labeled data $\mathcal{L} :< x_i, y_i >$ for i=1 to N.
- Find the best threshold θ^* at which fruits switch from being safe to noxious.
- Problem: We need a fast and cost effective approach, where the farmer do not have to harvest 100's of fruits and neither risk animals to fall ill.

Motivation: A simple and a cost effective solution

- Using PAC learning framework (Valiant, 1984), if underlying distribution is perfectly classified into some hypothesis h in the Hypothesis class $\mathcal{H}$ (i.e. set of all values for θ). We only need to test $\mathcal{O}(1/\epsilon)$ randomly selected fruits, where ϵ is the desired maximum error rate.
- Algorithm: Binary search
  - Arrange 9 fruits based on irregularity measure and apply binary search.
  - Find the threshold θ^* at which fruits switch from being safe to noxious.
  - The classifier with error ϵ or less can be found with a mere $\mathcal{O}(\log_2 1/\epsilon)$.
- Exponential reduction in the number of tests required for classification.
- Avoids the need to harvest hundreds of fruits and risk animal's health.

Active learning: Study

Brief Overview

- Introduction to active learning
- Minimising expected error
- Variance reduction
- Use of optimal designs
- Batch mode active learning
- Summary and discussions

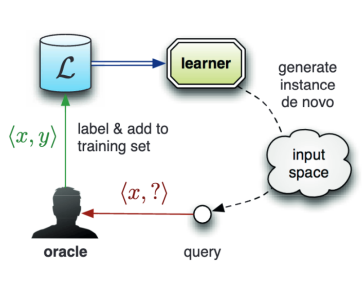
Introduction to active learning

- Traditional machine learning algorithms rely on existing training data to induce a hypothesis about the problem at hand. This process can be costly in terms of the amount of data required.
- Active learning, on the other hand, is a continuous interactive learning process that develops and tests new hypotheses as part of the learning process.
- Compared to traditional machine learning, active learning can learn tasks with less training data, making it a more efficient approach.
- Active learning is particularly useful for solving complex problems that have high-dimensional features and unreliable labels.
- However, implementing the active learning query framework can be more expensive than simply collecting a small number of labeled instances, which may be sufficient in some cases.

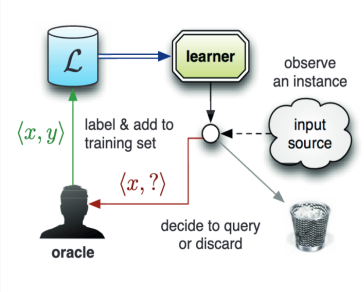
Active learning - How to begin with ?

- Assumptions:
  - Queries on instance labels are answered.
  - The appropriate hypothesis class for the problem is more or less already decided (e.g., naive Bayes, decision trees, neural networks, etc.).
- Query format:
  - Queries take the form of unlabelled instances.
  - The hypothesis class is known and fixed.
- Active learning assumes that these two assumptions hold, and the unlabelled instances can be used to reduce the cost of labelling while maintaining the same level of accuracy. The approach works best when there are numerous unlabelled instances that can be easily collected or synthesised.

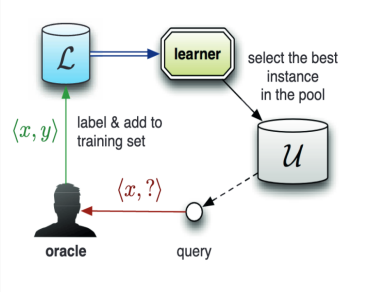
Active learning - different approaches



a) Query synthesis



b) Stream based



c) Pool based

The Problem of Selecting the "Best" Instance

- The task of an active learner is to identify the "best" instance to query.
- But what does "best" mean?.
- In previous sessions, we have explored heuristics based on uncertainty and hypothesis space selection.
- But what if we don't necessarily care about model certainty or correctness of hypotheses?.
- What we may really care about is how well the model makes predictions.

Decision Theory and Active Learning

- The problem with selecting the "best" instance is that we don't know the answer or our error before asking the question.
- Decision theory helps us make decisions under uncertainty by minimizing error as an expected value as we cannot reduce the error as a known value.
- We can identify all possible outcomes, determine their values and probabilities, and compute a weighted sum to give an expected value for each action.
- The "rational" decision is to choose the action that results in the lowest expected future error.
- In active learning, this means selecting the instance that is most likely to reduce future error once we know its answer.

Minimising expected error in Active Learning

- To minimize future error, the active learner needs to make decisions under uncertainty.
- Decision theory helps by computing expected values of future outcomes based on probabilities.
- To compute expected error, the learner needs two probability distributions:
  - Probability of the oracle's label y in answer to query x.
  - Probability of the learner making an error on some other instance $x^{'}$ once the answer is known.
- Both probabilities are not known, but can be approximated using the model's posterior distribution.
- If a large unlabeled pool $\mathcal{U}$ is available, the learner can minimize expected error over it, assuming it's representative of the test distribution.

Continued

- To minimise the expected classification error over the unlabeled data $\mathcal{U}$, the decision-theoretic utility measure is:

$$
\begin{aligned}
x^*_{ER} &= \underset{x}{\operatorname{argmin}}\, \mathbb{E}_{Y|\theta,x}\left[\sum_{x' \in \mathcal{U}} \mathbb{E}_{Y|\theta^+,x'}[y \neq \hat{y}]\right] \\
&= \underset{x}{\operatorname{argmin}} \sum_{y} P_\theta(y \mid x)\left[\sum_{x' \in \mathcal{U}} 1 - p_{\theta^+}(\hat{y} \mid x')\right].
\end{aligned}
$$

- The objective is to minimize the expected total number of incorrect predictions (excluding "near misses").

Continued

The expected log-loss by re-training the model with a new labeled set $\mathcal{L} \cup < x, y >$ which includes the candidate query x and the hypothetical oracle response y. The updated model is referred to as θ^+,

$$
\begin{aligned}
x^*_{LL} &= \underset{x}{\operatorname{argmin}}\, \mathbb{E}_{Y|\theta,x}\left[\sum_{x' \in \mathcal{U}} \mathbb{E}_{Y|\theta^+,x'}\left[-\log p_{\theta^+}\left(y \mid x'\right)\right]\right] \\
&= \underset{x}{\operatorname{argmin}} \sum_y P_\theta(y \mid x)\left[\sum_{x' \in \mathcal{U}} -\sum_{y'} p_{\theta^+}\left(y' \mid x'\right) \log p_{\theta^+}\left(y' \mid x'\right)\right] \\
&= \underset{x}{\operatorname{argmin}} \sum_y P_\theta(y \mid x) \sum_{x' \in \mathcal{U}} H_{\theta^+}\left(Y \mid x'\right),
\end{aligned}
$$

in is equivalent to the expected total future output entropy (uncertainty) over $\mathcal{U}$. The entropy of a probability distribution measures the amount of uncertainty or "surprise" associated with that distribution.

Continued

- $P_\theta(y \mid x)$ - probability of true label y given instance x, after fitting the model, predicted distribution of the response.
- An instance x from unlabeled pool $\mathcal{U}$, queried to the oracle and received label y, is added to the labeled pool $\mathcal{L}$ and then the model is re-trained. The updated parameters of the model are denoted as θ^+.
- $P_{\theta^+}(y^{'} \mid x^{'})$ - probability of assigning to label $y^{'}$ for given instance $x^{'}$, after updating the model.
- $H_{\theta^+}(Y \mid x')$, is the entropy over all possible labels of $y^{'}$ for a given instance $x^{'}$.

Expected Error Reduction for Text Classification

- Goal: Assign pre-defined categories to textual data.
- Model: Naive Bayes classification.
- Active Learning Strategies:
  - Density-weighted query by committee
  - Uncertainty sampling
  - Expected error reduction (based on log-loss)
- Results:
  - All active approaches $>$ random sampling.
  - Expected error reduction produces more accurate classifiers with less labeled data.
- Learning curves:
  - X-axis: number of labeled examples.
  - Y-axis: classification accuracy.
  - Expected error reduction outperforms other strategies.

Continued

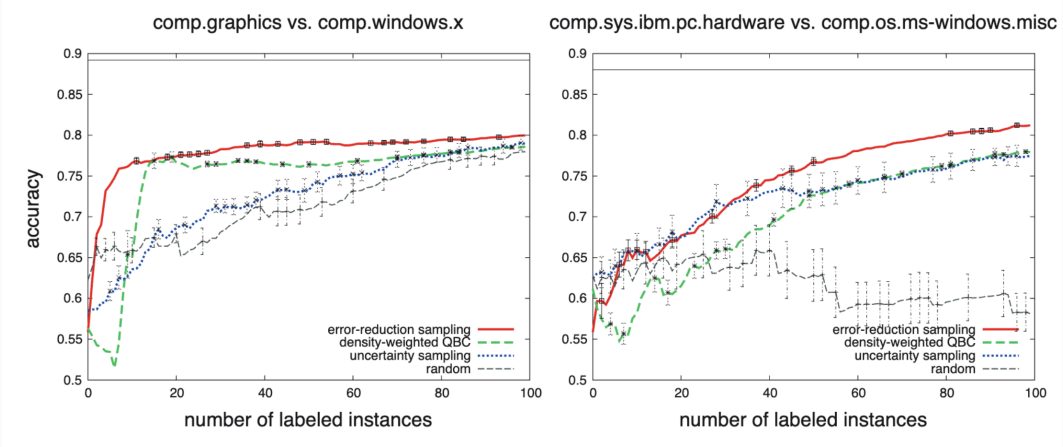


Figure 2: Learning curves for two binary classifications for different active learning approaches in comparison with random sampling. (Roy and McCallum, 2001).

Computational cost of expected error reduction

- Expected error reduction is a computationally expensive approach to active learning.
- It requires estimating the expected future error over the unlabeled pool for each query.
- A new model must be re-trained for every possible labeling of every possible query in the pool.
- For example, a binary logistic regression model would require $\mathcal{O}(|\mathcal{U}||\mathcal{L}||\mathcal{G}|)$ time simply to choose the next query, where $\mathcal{U}$ is the size of the unlabeled pool, $|\mathcal{L}|$ is the size of the current training set, and $|\mathcal{G}|$ is the number of gradient computations required by the optimization procedure until convergence.

Estimated Error Reduction Framework and limitations

- Framework is near-optimal and not dependent on model class.
- Requirements: appropriate objective function and estimation of posterior label probabilities.
- Strategies used with a variety of models: naive Bayes, Gaussian random fields, logistic regression, and support vector machines and Can optimize any performance measure, such as precision, recall, F1-measure, or area under the ROC curve.
- Expected error reduction is a computationally expensive approach to active learning as it requires estimating the expected future error over the unlabeled pool for each query.
- Expected error reduction is mostly used in simple binary classification tasks.

Indirect Error Reduction - Variance reduction

- Directly minimizing an error function is costly and generally cannot be done in closed form.
- Model must be re-trained using hypothetical labelings to estimate expected reduction in error.
- This can be computationally expensive.
- However, in some cases, we can reduce generalization error indirectly by minimizing output variance.
- This approach sometimes has a closed-form solution.

Variance reduction - derivation

Bias and variance decomposition (Geman et al., 1992)

$$\begin{aligned}\mathbb{E}\left[(\hat{y}-y)^2 \mid x\right] &= \underbrace{\mathbb{E}_{Y|x}\left[\left(y-\mathbb{E}_{Y|x}[y \mid x]\right)^2\right]}_{\text{noise}} \\ &+ \underbrace{\left(\mathbb{E}_{\mathcal{L}}[\hat{y}]-\mathbb{E}_{Y|x}[y \mid x]\right)^2}_{\text{bias}} \\ &+ \underbrace{\mathbb{E}_{\mathcal{L}}\left[\left(\hat{y}-\mathbb{E}_{\mathcal{L}}[\hat{y}]\right)^2\right]}_{\text{output variance}}\end{aligned}$$

- $\mathbb{E}_{Y|x}[.]$ represents expectation over $P(y \mid x)$,
- $\mathbb{E}_{\mathcal{L}}[.]$ represents expectation over the labeled training set $\mathcal{L}$,
- $\mathbb{E}[.]$ represents expectation over both $P(y \mid x)$ and $\mathcal{L}$,
- $\hat{y}$ is the model's prediction for a given instance x,
- y is the true label for that instance.

Decomposing Generalization Error

- Generalization error decomposes into three components:
  - Noise: unreliability of true label y given x, which does not depend on model or training data.
  - Bias: error due to the model class itself.
  - Output variance: remaining component of the learner's squared-loss with respect to the target function.
- Minimizing noise and bias is challenging, but minimizing output variance is guaranteed to minimize future generalization error.
- Minimizing output variance can be accomplished by reducing model complexity or using regularization techniques.

Reducing Squared-Loss Error through Efficient Instance Labeling

- Goal: Reduce error in squared-loss sense by labeling instances that decrease model's output variance over unlabeled instances,

$$x^*_{VR} = \underset{x}{\operatorname{argmin}} \sum_{x' \in \mathcal{U}} \mathrm{Var}_{\theta^+}\left(Y \mid x'\right)$$

- θ^+ denotes the model after re-training with $\mathcal{L} \cup < x, y >$.
- Challenge: How to compute this value more efficiently than direct error minimization through re-training .

Optimal Experimental Design and Fisher Information in Active Learning

- Optimal experimental design: approaches to derive closed form utility functions for active learning.
- Focus on regression tasks with few input variables as predictors.
- Key ingredient: Fisher information.
- Fisher information: measures how much information a random variable Y carries about a parameter θ of the likelihood function $P_\theta(Y)$
- Fisher score: partial derivative of the logarithm of $P_\theta(Y)$ with respect to θ:

$$\nabla x = \frac{\partial}{\partial \theta} \log P_\theta(Y \mid x)$$

  where ∇x score of input instance x on which Y depends.
- Fisher information can be used to quantify the amount of information gained by labeling a particular instance in active learning.

Fisher Information: Score and Variance

- The Fisher score (or gradient) is not dependent on the actual label of x, but only on the distribution over Y under parameters θ.
- For multiple parameters, the score is a vector:

$$\begin{aligned} F &= \mathbb{E}_X \left[\left(\frac{\partial}{\partial \theta} \log P_\theta(Y \mid x) \right)^2 \right] \\ &= \mathbb{E}_X \left[\frac{\partial^2}{\partial \theta^2} \log P_\theta(Y \mid x) \right] \\ &\propto \sum_x \nabla x \nabla x^\top \end{aligned}$$

- The Fisher information F is the variance of the Fisher score.
- Alternatively, Fisher score and information could be written as $\nabla_\theta x$ and $F(\theta)$ respectively, to make their relationship with model parameters explicit.
- Fisher information is not a function of a particular observation, as we integrate over all instances in a particular input distribution.

Optimizing Fisher Information: D, E, and A-optimality

- Active learners should select data that maximizes the Fisher information (or minimizes its inverse) to minimize parameter estimate variance.
- For models with K parameters, Fisher information takes the form of a K x K covariance matrix, making optimization less clear.
- Two main optimal experimental designs:
  - D-optimality: minimizes determinant of the inverse information matrix
  - A-optimality: minimizes trace of the inverse information matrix, resulting in minimizing average variance of parameter estimates

Optimizing Fisher Information: D - optimality

- D-optimality relates to minimizing differential posterior entropy of parameter estimates

$$x_D^* = \arg\min_x \mathbf{det}\left(\left[F_{\mathcal{L}} + \nabla x \nabla x^\top\right]^{-1}\right)$$

  where additive property of Fisher Information, add information of x to all previous training observations in $\mathcal{L}$ (Chaloner and Verdinelli, 1995).
- D-optimal design criterion provides a closed-form solution, without actual model re-training.
- Determinant can be thought of as a measure of volume.
- D-optimal design aims to select instances that reduce the amount of uncertainty in parameter estimates.

A-Optimal Design

- A-optimal designs aim to reduce the average variance of parameter estimates by focusing on values along the diagonal of the information matrix.
- A common variant of A-optimal design is to minimize $\mathbf{tr}\left(A\ F_{\mathcal{L}}^{-1}\right)$, where A is a square, symmetric "reference" matrix.
- A matrix of rank one can be used as a special case, in particular: $A_x = \nabla x\ \nabla x^T$.
- In this case, $\mathbf{tr}\left(A_x\ F_{\mathcal{L}}^{-1}\right) = \nabla x\ F_{\mathcal{L}}^{-1}\ \nabla x^T$, which is the equation for computing the output variance for a single instance x in regression models (Schervish, 1995).
- A-optimal designs attempt to minimize the prediction variance for all data instances.

A-optimal design in Active learning

- Using A-optimal design, we can derive the utility measure for active learning, fisher information ratio:

$$
\begin{aligned}
x^*_{FIR} &= \underset{x}{\operatorname{argmin}} \sum_{x' \in \mathcal{U}} \mathrm{Var}_{\theta^+}\left(Y \mid x'\right) \\
&= \underset{x}{\operatorname{argmin}} \sum_{x' \in \mathcal{U}} \mathbf{tr}\left(A_{x'}\left[F_{\mathcal{L}} + \nabla x \nabla x^\top\right]^{-1}\right). \\
&= \underset{x}{\operatorname{argmin}}\, \mathbf{tr}\left(F_{\mathcal{U}}\left[F_{\mathcal{L}} + \nabla x \nabla x^\top\right]^{-1}\right)
\end{aligned}
$$

  where $\mathrm{Var}_{\theta^+}(.)$ denotes the variance of re-trained model with query x and its label y.

- Adding the information of x to current training set $\mathcal{L}$, without explicit re-training the model, we can estimate in a closed-form.
- The utility measure is the inner product of two matrices as the matrix traces are additive.

A-optimal design in Active learning (continued)

- The Fisher information $F_{\mathcal{U}} = \sum_{x' \in \mathcal{U}} A_{x'}$ represents the model's output variance across the input distribution $\mathcal{U}$, which cannot be explained by the observations in $\mathcal{L} \cup \{x\}$.
- The ratio of Fisher information matrices can be used as a utility measure for active learning, called the Fisher information ratio.
- Querying the instance which minimizes this ratio indirectly reduces generalization error (with respect to $\mathcal{U}$) in the squared-loss sense.
- This approach has an advantage over explicit error reduction as it does not require retraining the model for each possible labeling of each candidate query.
- The information matrices provide an approximation of the updated output variance, simulating retraining without actual retraining.

Drawbacks of Variance-Reduction Methods

- Estimating output variance requires inverting and multiplying large matrices, making it computationally complex.
- Operations require $\mathcal{O}(K^3)$ time, where K is the number of parameters in the model.
- Computational complexity remains $\mathcal{O}(U\ K^3)$ for selecting the next query, as operations must be repeated for every instance in $\mathcal{U}$ being considered for querying.
- Few experimental comparisons with other active learning methods, and existing studies have reported mixed results.
- Heuristics such as dimensionality reduction, matrix approximation, and sampling have been proposed to reduce computational complexity, but are still orders of magnitude slower than simpler query strategies like uncertainty sampling.

Submodular Functions in Batch-Mode Active Learning

- Active learning typically involves selecting queries in a serial manner.
- Batch-mode active learning selects queries in groups, making it more suitable for parallel labeling environments or slow training procedures.
- Active learning heuristics based on variance reduction naturally lend themselves to the batch setting with performance guarantees.
- Submodularity is a property of set functions that intuitively formalizes the idea of diminishing returns.
- Adding a instance x to set $\mathcal{A}$ has more gain in its utility function, rather than adding x to a larger set $\mathcal{A}'$, where $\mathcal{A} \subseteq \mathcal{A}'$. A set function s is submodular if:

$$s(\mathcal{A} \cup \{x\}) - s(\mathcal{A}) \geq s\left(\mathcal{A}' \cup \{x\}\right) - s\left(\mathcal{A}'\right)$$

or

$$s(\mathcal{A}) + s(\mathcal{B}) \geq s(\mathcal{A} \cup \mathcal{B}) + s(\mathcal{A} \cap \mathcal{B})$$

Using Submodular Functions for Variance Reduction

- Variance reduction heuristics can be recast as monotonically non-decreasing functions, allowing for submodularity and performance guarantees in the active learning setting.
- A variance reduction heuristic can be recast as a monotonically non-decreasing function by measuring the total difference between the model's output variance before querying the set $\mathcal{Q}$ and the expected variance afterward.

$$\begin{aligned} s(\mathcal{Q}) &= \sum_{x \in \mathcal{U}} \mathrm{Var}_{\theta}(Y \mid x) - \mathrm{Var}_{\theta^{+}\mathcal{Q}}(Y \mid x) \\ &= \mathrm{tr}\left(F_{\mathcal{U}} F_{\mathcal{L}}^{-1}\right) - \mathrm{tr}\left(F_{\mathcal{U}}\left[F_{\mathcal{L}} + F_{\mathcal{Q}}\right]^{-1}\right) \end{aligned}$$

- Greedy algorithms for submodular criteria can be used for certain learning algorithms such as Gaussian processes, logistic regression, and linear regression.
- In settings where there is a fixed budget for gathering data, submodular utility measures guarantee near-optimal results with significantly less computational effort.

Advantages of Submodular Functions and limitations

- Greedy algorithms for selecting N instances using submodular functions provide a performance guarantee of $(1 - 1/e) * s(Q^*_N)$, where $s(Q^*_N)$ is the value of the optimal set of size N.
- Monotonically non-decreasing submodular functions guarantee a lower-bound performance of around 63% of optimal.
- However, it is worth noting that set optimization problems, such as those involving submodular functions, are generally NP-hard, which means that the computations required to solve them can scale exponentially with the size of the input.

Difference between Expected error and variance reductions

Expected error reduction

- Focuses on minimizing the expected classification error by selecting the unlabeled data points that are most uncertain or informative.
- Assumes that the data follows a fixed distribution, and the goal is to obtain a model that performs well on the whole distribution.
- It can be more effective when the decision boundary is complex and uncertain, and there is a high degree of noise or overlap between classes.

Variance reduction

- Aims to reduce the variance of the classifier by selecting the data points that are most representative or diverse.
- Assumes that the data distribution is not fixed, and the goal is to obtain a model that is robust to changes in the data distribution.
- It can be more effective when the data distribution is non-stationary or the labeled data is highly biased.

Summary

- Discussed principled active learning strategies for minimizing expected error or output variance on unlabeled instances
- Methods for minimizing classification error by 0/1-loss or log-loss require re-training models for all possible labelings of queries
- Methods for minimizing output variance in squared-loss can be computed in closed form and are applicable to classification and regression problems
- These methods can be generalized for active learning in batches with efficiency guarantees based on submodularity
- Two major drawbacks are limited applicability to certain hypothesis classes and computational complexity for large problem sizes

References

- Burr Settles. Active Learning. CA: Morgan Claypool Publishers, 2012.
- L.G. Valiant. A theory of the learnable. Communications of the ACM, 27(11):1134-1142, 1984.
- N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In Proceedings of the International Conference on Machine Learning (ICML), pages 441-448. Morgan Kaufmann, 2001.
- S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. Neural Computation, 4:1-58, 1992.
- K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. Statistical Science, 10(3): 237-304, 1995.

Thank you!